# A Multi-objective Evolutionary Algorithm with New Reproduction and Decomposition Mechanisms for the Multi-Point Dynamic Aggregation Problem

Guanqiang Gao
stef_leon_gao@outlook.com
Beijing Institute of Technology
Beijing, China

Bin Xin
brucebin@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Yi Mei
yi.mei@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Shengyu Lu
lu_shengyu@126.com
Beijing Institute of Technology
Beijing, China

Shuxin Ding
dingshuxin@rails.cn
Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited
Beijing, China

## ABSTRACT

An emerging optimisation problem from real-world applications, named the multi-point dynamic aggregation (MPDA) problem, has become an active research of the multi-robot system. This paper focuses on a multi-objective MPDA (MO-MPDA) problem which is to design execution plans of robots for minimising the cost of used robots and maximising the efficiency of task execution. The MO-MPDA problem has the issues of conflicting objectives, redundant representation, and variable-length encoding, posing extra challenges to address the MO-MPDA problem effectively. Combining the $\varepsilon$-constraint method and decomposition mechanisms, a novel multi-objective evolutionary algorithm is proposed. The proposed algorithm selects the efficiency objective as the main objective and converts the cost objective as constraints. Thus, the multi-objective problem is decomposed into a series of scalar constrained optimisation subproblems by assigning each subproblem with an upper bound constraint. All the subproblems are optimised and evolved simultaneously with the transferring knowledge from other subproblems to solve the MO-MPDA problem parallelly and efficiently. Besides, considering the characteristics of parent individuals, this paper designs a hybrid reproduction mechanism to transmit effective information to offspring individuals for tackling the encoding redundancy and varying-length. Experimental results show that the proposed algorithm significantly outperforms the state-of-the-art algorithms in terms of most-used metrics.

## CCS CONCEPTS

• **Computing methodologies** → Robotics.


*GECCO '22, July 9–13, 2022, Boston, MA, USA*

ACM ISBN 978-1-4503-9237-2/22/07…$15.00
https://doi.org/10.1145/3512290.3528728

## KEYWORDS

Multi-objective evolutionary algorithm, hybrid reprodcution mechanism, multi-robot system, multi-point dynamic aggregation

**ACM Reference Format:**
Guanqiang Gao, Bin Xin, Yi Mei, Shengyu Lu, and Shuxin Ding. 2022. A Multi-objective Evolutionary Algorithm with New Reproduction and Decomposition Mechanisms for the Multi-Point Dynamic Aggregation Problem. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528728

## 1 INTRODUCTION

The Multi-Point Dynamic Aggregation (MPDA) problem is a task planning problem of the multi-robot system, which comes from the real world [13, 25, 28, 36]. Recently, it has become one of the active research topics due to its applications such as bushfire elimination, search and rescue, and medical resource scheduling domains [3, 7, 8, 18, 29]. Unlike the majority of scheduling and routing problems [1, 26, 31, 33], the demand of each task in the MPDA problem is time-varying and time-sensitive. Besides, multiple robots can execute one task simultaneously to complete the task efficiently. These distinctive characteristics lead to complex interactions among robots and tasks, and it is difficult for a decision-maker to design high-quality execution plans for robots to complete geographically distributed tasks.

To the best of our knowledge, the previous researches about the MDPA problem only focused on one single objective [9, 11, 30, 35], such as the maximal completion time and the total execution time of all the tasks. However, many real-world applications always considers two or more potentially conflicting objectives e.g., minimizing the cost of used robots and maximizing the efficiency of task execution, simultaneously. In this paper, we focus on the multi-objective MPDA (MO-MPDA) problem with the commonly considered objectives. An example of the MO-MPDA problem is shown in Fig. 1, which has some tasks (e.g. fire points) with time-varying demands and a depot with several robots in the mission environment. Fig. 1 uses four different colored segments, indicating four routes of robots to execute all tasks. In this paper, the cost objective is regarded as

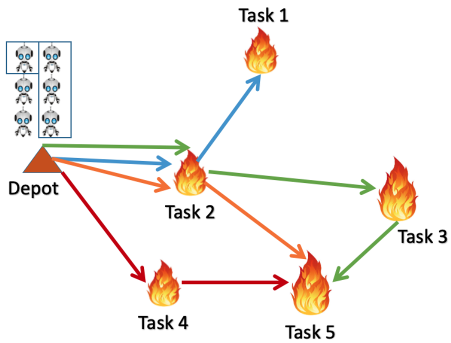


**Figure 1: An example of the MO-MPDA problem.**

the number of used robots, and the efficiency objective is regarded as the makespan (e.g. the maximal completion time of all the tasks).

The MO-MPDA problem has the following challenges. First, due to the multi-objective characteristic, the previous single-objective MPDA optimisation approaches [11, 13, 30] cannot efficiently handle the two conflicting objectives. The whole single-objective approach needs to be adjusted to satisfy the changing robot number. Second, the execution plan in the MO-MPDA problem is hard to represent since unfixed length of an execution plan and the number of used robots. Thus, the number of decision variables in two different individuals might be quite different in the decision space, which raises great difficulty solving MO-MPDA. Third, homogeneous robots in MO-MPDA leads to many isomorphic execution plans. Last but not least, similar to the single-objective MPDA problem, the time-varying demand and collaboration behaviours lead to a huge and complex solution space of MO-MPDA. It is hard to find a high-quality solution in the solution space.

The decomposition-based multi-objective evolutionary algorithm with the $\varepsilon$-constraint (DMOEA-$\varepsilon$C) simultaneously optimizes constrained optimization subproblems with different upper bounds [4]. DMOEA-$\varepsilon$C is a very competitive method for addressing multi-objective problems. Especially it shows obvious advantages on combinational optimization problems [4, 5, 21, 22]. Since the MO-MPDA problem is a combinational optimisation problem with tight constraints, we expect DMOEA-$\varepsilon$C to be effective in addressing the MO-MPDA problem.

The overall goal of this paper is to develop an effective algorithm to obtain Pareto solutions of MO-MDPA for a decision-maker. To achieve this goal, research contributions are shown as follows specifically.

- We formulate a MO-MPDA problem with two conflicting objectives: to minimize the number of used robots and the makespan.
- An encoding method using matrices of different shapes and a decoding method distinguishing arrival and departure events are proposed in this paper. In a matrix encoding, each row represents the task-executing sequences for a robot, and the number of rows represents the number of used robots.
- A novel offspring reproduction mechanism is proposed in this paper. To tackle the representation redundancy, all visiting sequences of robots are first sorted according to the characteristics of tasks in the proposed reproduction mechanism. Then, three crossover operators and one mutation operator are designed to generate offspring individuals for effective information transmission from parent individuals.
- The framework and mechanisms (i.e. offspring reproduction and subproblem-to-individual matching mechanisms) of DMOEA-$\varepsilon$C are re-designed to fit the characteristics of the MO-MPDA problem. The proposed algorithm selects the makespan as the main objective and converts the number of used robots as a scalar of constraints.

The rest of this paper is organised as follows. The mathematical model of the MO-MPDA problem are presented in Section II. Then, Section III describes the proposed DMOEA-$\varepsilon$C method. Sections IV and V present the experimental results and performance analysis of the proposed mechanisms. Finally, this paper is concluded in Section VI.

# 2 BACKGROUND

## 2.1 Problem Description

There are a depot and a number of tasks (e.g. fire points and disaster victims) with time-varying demands in the MO-MPDA scenario. We use an undigraph $G(\mathcal{V}, \mathcal{E})$ to define the MO-MPDA problem. In the set of vertexes $\mathcal{V} = \{v_1, \dots, v_{N+1}\}$, $\{v_1, \dots, v_N\}$ indicates the set of tasks, and $v_{N+1}$ indicates the depot. Each task $v_i$ has an inherent time-varying demand $q_i(t)$, which changes based on the following equation

$$q_i(t) = q_i(0) + \alpha_i \times t, \tag{1}$$

where $\alpha_i$ represents the inherent increment rate of task $v_i$. In the set of edges of $G$, every edge indicates a route among two tasks with the travel time $t_{i,j}$.

There are $\omega$ homogeneous robots located at the depot, which will execute and complete all the time-varying tasks. Every robot has the same ability $\beta$, representing the amount of demand which it can reduce per time unit. Fig. 2 shows an example demonstrating the relationship between the task demand and abilities of robots. In the figure, the task's demand is decreased to 0 about time 12, and the task is completed by three robots $rob_1, rob_2$, and $rob_3$.

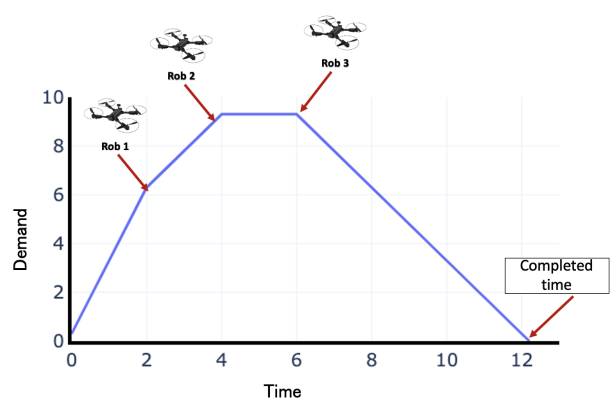


**Figure 2: An example of a task's demand executed by three robots over time.**

*2.1.1 Mathematical model.* The over goal of MO-MPDA is to design a series of Pareto execution plans for a decision-maker, which makes all tasks completed as soon as possible with as few robots as possible. Based on the aforementioned descriptions, the MO-MPDA mathematical model can be defined as follows.

$$\min f_1 = \max_{i \in \{1,2,\cdots,N\}} ct_i \tag{2}$$

$$\min f_2 = K \tag{3}$$

$$s.t. \sum_{i=1}^{N+1} \sum_{k=1}^{K} x_{i,j}^k \geq 1, \forall j \in \{1,2,\ldots,N\} \tag{4}$$

$$\sum_{j=1}^{N+1} x_{i,j}^k = \sum_{j=1}^{N+1} x_{j,i}^k, \quad \forall i \in \{1,2,\ldots,N+1\}, \forall k \in \{1,2,\ldots,K\} \tag{5}$$

$$\sum_{i=1}^{N+1} x_{i,j}^k \leq 1, \quad \forall j \in \{1,2,\ldots,N\}, \forall k \in \{1,2,\cdots,K\} \tag{6}$$

$$at_{N+1,k} = ct_{N+1} = 0, \forall k \in \{1,2,\cdots,K\} \tag{7}$$

$$at_{j,k} = \sum_{i=1}^{N+1} (ct_i + t_{i,j}) x_{i,j}^k, \quad \forall j \in \{1,2,\ldots,N\}, \forall k \in \{1,2,\cdots,K\} \tag{8}$$

$$q_j(0) + \alpha_j ct_j = \sum_{i=1}^{N+1} \sum_{k=1}^{K} x_{i,j}^k \beta (ct_j - at_{j,k}), \quad \forall j \in \{1,2,\ldots,N\} \tag{9}$$

$$\alpha_j < \sum_{k=1}^{K} \sum_{i=1}^{N+1} x_{i,j}^k \beta, \forall j \in \{1,2,\ldots,N\} \tag{10}$$

$$K \leq \omega \tag{11}$$

$$x_{i,j}^k \in \{0,1\}, i \neq j, \quad \forall i,j \in \{1,2,\ldots,N+1\}, \forall k \in \{1,2,\ldots,K\} \tag{12}$$

where the integer decision variable $K$ represents the number of used robots to execute all the tasks, and the binary decision variables $x_{i,j}^k$ takes 1 if $rob_k$ travels from $v_i$ to $v_j$, and 0 otherwise. $at_{j,k}$ and $ct_j$ are auxiliary variables to describe MO-MPDA. $at_{j,k}$ represents the arrival time of $rob_k$ at $v_j$, and $ct_j$ represents the completion time of $v_j$ respectively.

One objective (2) of the MO-MPDA problem is to minimize the maximum completion time of all tasks, and the other objective (3) is to minimize the number of robots executing all tasks. Constraint (4) ensures that each task is executed by at least one robot. Constraint (5) indicates that the number of outgoing routes equals the number of incoming routes for each task and each robot. Constraint (6) ensures that each task is executed by each robot at most once. Constraint (7) sets the arrival time and completion time for the depot to 0. Constraint (8) specifies the relationship between $ct_j$, $at_{j,k}$ and $x_{i,j}^k$. Constraint (9) implies that a task is completed when its demand decreases to zero (i.e. the accumulated demand from time 0 to $ct_j$ equals the total demand reduced by the robots executing the task during this time period). It also shows the time-varying characteristic of the task demand. Constraint (10) indicates that for each task, the total ability of the robots executing it must be greater than its inherent increment rate. Otherwise, the task can never be completed. Constraint (11) indicates the number of assigned robots is no greater than the number of robots in the depot. Constraint (12) sets the binary domain of the decision variables.

*2.1.2 Bound determination.* There is a scope of the cost of used robots for a specific MO-MPDA problem. Then, the low bound and upper bound of the number of used robots can be calculated by Eqs. (13) and (14).

$$K_{lb} = \lceil \frac{\max_{i \in \{1,\cdots,N\}} \alpha_i}{\beta} \rceil \tag{13}$$

$$K_{ub} = \sum_{i=1}^{N} \lceil \frac{\alpha_i}{\beta} \rceil \tag{14}$$

where $\lceil\rceil$ represents a ceiling function, $K_{lb}$ indicates the minimal number of used robots which ensures every task can be completed, and $K_{ub}$ indicates the maximal number of used robots which ensures all the tasks can be executed by one assignment.

## 2.2 Related Work

*2.2.1 MPDA.* There are several approaches proposed to address the MPDA problem. Hao *et al.* proposed an evolutionary computation method hybrid with differential evolution and estimation of distribution algorithm [14]. Comparison experiments showed that the hybrid method outperforms the differential evolution in terms of the convergence speed and solution quality. Xin *et al.* proposed an estimation of distribution algorithm with two different probability models to address the MPDA problem [35]. The method [35] outperforms the genetic algorithm due to the two probability models. Gao *et al.* proposed a genetic programming hyper-heuristic method to evolve the execution rules of robots in MPDA. Experiments showed that the automatic evolved rules significantly achieved a better performance than the state-of-the-art manually designed rules.

*2.2.2 DMOEA-$\varepsilon$C.* Zhang *et al.* firstly proposed a multi-objective evolutionary algorithm based on decomposition (MOEA/D) in 2007 [37]. The algorithm shows very promising results for approximating the Pareto front. The main idea of MOEA/D is the decomposition mechanism, the method decomposes a multi-objective optimisation problem into a number of scalar optimisation subproblems[2, 20, 34]. Each subproblem learns valid information from its neighbouring subproblems so that all the subproblems are evolved parallel and efficiently. Thank to the collaborative optimisation mechanism, the MOEA/D method has a fast computation speed. Since a scalar of weight vectors uniformly splits the solution space, the MOEA/D method also has a good population diversity.

DMOEA-$\varepsilon$C belonging to MOEA/D firstly combined MOEA/D methods and the $\varepsilon$-constraint to address a multi-objective problem [4]. In DMOEA-$\varepsilon$C, a multi-objective problem is decomposed into several constrained optimisation subproblems with different upper bounds. The DMOEA-$\varepsilon$C method simultaneously optimises these constrained optimisation subproblems using their neighbour

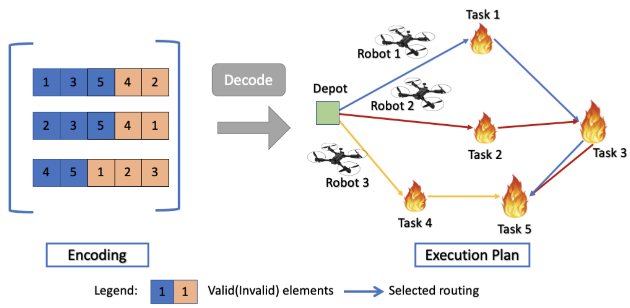


**Figure 3: An example of the decoding processes with a given encoding for MO-MPDA.**

information. The details of DMOEA-$\varepsilon$C are shown as follows[5].

$$\begin{aligned} &\text{minimize} \quad f_{main} = f_s(\mathbf{x}) \quad + \rho \sum_{i=1}^{m} f_i(\mathbf{x}) \\ &\text{subject to} \begin{cases} \frac{f_i(\mathbf{x}) - z_i^*}{z_i^{nad} - z_i^*} \le \varepsilon_i, \forall i \in \{1, 2, \cdots, m\} / \{s\} \\ \mathbf{x} = (x_1, x_2, ..., x_n) \in \Omega \end{cases} \end{aligned} \tag{15}$$

where $s$ represents the predefined main objective index, $0 \le \varepsilon = (\varepsilon_1, ..., \varepsilon_{s-1}, \varepsilon_{s+1}, ..., \varepsilon_m) \le 1$ is the upper bound vector, $\rho$ is a small positive number, $\mathbf{z}^* = (z_1^*, ..., z_m^*)$ and $\mathbf{z}^{\mathbf{nad}} = (z_1^{nad}, ..., z_m^{nad})$ are the ideal point and the nadir point, respectively.

# 3 THE PROPOSED DMOEA-$\varepsilon$C ALGORITHM

## 3.1 Encoding and Decoding

*3.1.1 Encoding.* The explicit representation of the execution plan of all the robots in the MPDA problem is a variable-length sequence of events. To simplify the representation, an implicit representation of a solution for the MO-MPDA problem is adopted in this paper, which is a matrix and shown in (16):

$$X = \begin{bmatrix} \pi_{1,[1]} & \pi_{1,[2]} & \cdots & \pi_{1,[N]} \\ \pi_{2,[1]} & \pi_{2,[2]} & \cdots & \pi_{2,[N]} \\ \vdots & \vdots & \vdots & \vdots \\ \pi_{m,[1]} & \pi_{m,[2]} & \cdots & \pi_{m,[N]} \end{bmatrix}. \tag{16}$$

Each row of the given matrix has $N$ integral elements which is a permutation of all tasks' indexes. Similar to the representation in VRPs [24, 32], the elements of one row indicate the task-executing sequences. For example, if $i$th row is $[2, 3, 1]$, $rob_i$ will intend to execute tasks $v_2$, $v_3$ and $v_1$ in order.

*3.1.2 Decoding.* The decoding method used in this paper adopts the event trigger mechanism. When a robot completes a task, it becomes active and selects its next executing task according to the corresponding encoding. The details of the decoding method are similar to the decoding process in [11, 12]. Fig. 3 shows a 3-robot-5-task example of the decoding process of the MO-MPDA problem with a given encoding. $v_5$ which is executed by three robots simultaneously is the last completed task. Thus, for the MO-MPDA problem, the first objective of the example solution in Fig. 3 is the completion time of $v_5$, and the second objective is 3.

Some elements in a given encoding is redundant since a robot usually does not visit all tasks. In this paper, the number of invalid elements in a row is denoted as $ie_k$. For example, $ie_3$ of $rob_3$ is 3 in Fig. 3. It also should be noticed that many encoding representations have the same objective values as robots are homogeneous. For example, swapping the visiting sequences of $rob_1$ and $rob_2$ in Fig. 3 does not affect the two objective values.

## 3.2 Framework

In this paper, the new DMOEA-$\varepsilon$C method proposed for MO-MPDA selects the makespan objective as the main objective, and converts the number of used robots as the constraints. The formulation details are shown as follows.

$$\begin{aligned} &\text{minimize} \quad f_{main} = f_1(\mathbf{x}) + \rho f_2(\mathbf{x}) \\ &\text{subject to} \begin{cases} \frac{f_2(\mathbf{x}) - \mathrm{K_{lb}}}{K_{ub} - K_{lb}} \le \varepsilon_i \\ \mathbf{x} \in \Omega \end{cases} \end{aligned} \tag{17}$$

where the calculation methods for $f_1$ and $f_2$ are shown in Eqs (2) and (3).

The framework of the proposed DMOEA-$\varepsilon$C method is shown in Algorithm 1. DMOEA-$\varepsilon$C contains four main components, 1) initialisation, 2) reproduction, 3) matching, and 4) Pareto updating. In the method, $R$ upper bounds for the number of used robots are maintained for the evolutionary process, where $R$ is the population size. At the beginning of DMOEA-$\varepsilon$C, $R$ solutions are randomly initialised. Then, the rest three components of DMOEA-$\varepsilon$C are run iteratively until the maximal number of fitness evaluations is reached. In each generation of the proposed method, $|I|$ individuals are selected to reproduce new individuals $\mathbf{Y}$ firstly. Second, each of the newly generated individuals $\mathbf{Y}$ is matched to a subproblem using the proposed matching method. Finally, the external archive **EP** is updated according to the current population.

## 3.3 Initialisation

After the range for the number of used robots is calculated, a series of scalar upper bounds $\varepsilon = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_R)$ are generated by a fixed interval $\triangle = 1/(R-1)$. Accordingly, the multi-objective problem can be decomposed into $R$ subproblems. The formulation for a subproblem $sp_i$ in DMOEA-$\varepsilon$C is shown as Eq. (17).

Due the integer characteristic of the robot number, it is obvious that the optimal number of robots for a subproblem $sp_i$ is $\lfloor \varepsilon_i (K_{ub} - K_{lb}) + K_{lb} \rfloor$, denoted by $m_i$. Thus, an initial individual with $m_i$ robots for a subproblem $sp_i$ is generated randomly. It should be noticed that different subproblems may have the same optimal number of robots. After initialising the individuals, DMOEA-$\varepsilon$C determines the neighbourhood $B(i)$ for each subproblem and sets the same utility value of subproblems for dynamic resource allocation [38]. At the end of initialisation component, the number of fitness evaluations and generation are set to $R$ and 0 respectively.

## 3.4 Reproduction

A new reproduction mechanism is designed in this paper to generate diversified offspring for propagating useful information. Recall that each individual is represented as multiple permutations of the tasks' indexes, and previous researchers have developed a number of crossover operators and mutation operators [16, 27]. The proposed reproduction mechanism is based on the classical crossover operator (i.e. partially matched crossover) and mutation operator

**Algorithm 1:** The proposed DMOEA-$\epsilon$C method

**Input:** A MO-MPDA instance, related parameters.
**Output:** An external archive population **EP**.

Calculate the upper and low bounds for the number of used robots, and generate $R$ evenly spread upper bound vectors;
According to the upper and low bounds and spread bound vectors, initialise the evolving population $\mathbf{Pop} = \{\mathbf{x}^1, \mathbf{x}^2 \ldots, \mathbf{x}^R\}$ randomly;
Evaluate each individual in **Pop**;
Extract nondominated individuals from **Pop** denoted as **EP**;
**for** $i = 1 \rightarrow R$ **do**
  Determine the neighbourhood $B(i)$ of the $i$th subproblem;
  Set the utility value $\Pi^i$ of the $i$th subproblem for the dynamic resource allocation;
**end**
$gen = 0, n = R$;
**if** *gen is a multiple of* $DRA\_interval$ **then**
  Update the indices of the subproblem $I$ that will be evolved in the next generations using the dynamic resource allocation mechainism in [38].
**end**
**while** $n \le NFE$ **do**
  **for** $i \in I$ **do**
    $P = \begin{cases} B(i), & \text{if } rand() < \delta \\ \{1, 2, ..., R\}, & \text{otherwise} \end{cases}$
    Use the proposed reproduction mechanism to generate offspring individuals **Y** from parent individuals $P$ ( **Algorithm** 2 ) ;
    **for** $Y \in \mathbf{Y}$ **do**
      **if** *$Y$ is infeasible* **then**
        Repair $Y$;
      **end**
      Evaluate the new individual $Y$, $n = n + 1$;
      Find the suitable subproblem $sp_{i^*}$ for the new individual $Y$ using the proposed mathcing mechanism (**Algorithm** 3);
      Compare $Y$ with the individual of the subproblem $sp_{i^*}$, and update individual and neighbouring individuals of the subproblem $sp_{i^*}$.
      Update the external archive **EP** and utility values $\Pi$ ;
    **end**
  **end**
  $gen = gen + 1$;
**end**
**return EP**

(i.e. swap mutation) [17]. The details of the proposed reproduction mechanism are shown in Algorithm 2.

In the reproduction mechanism, there are two different generation operators for offspring individuals. When a sampled value is less than $\delta_g$, the new offspring individuals are generated by the designed crossover operator. Otherwise, the new offspring individuals are generated by the designed mutation operator.

Since the robots in the MO-MPDA problem are homogeneous, the permutation encoding has a certain redundancy. At the beginning of the designed crossover operator, permutations in each parent individual are sorted according to the number of invalid elements and task indexes to address the redundancy issue. Then, the two

**Algorithm 2:** The new reproduction mechanism

**Input:** Parents individuals $X_a$ and $X_b$.
**Output:** New generated offspring individuals **Y**.

**if** $rand < \delta_g$ **then**
  Sort $X_a$ and $X_b$ based on the number of invalid elements and task indexes from small to large;
  **if** *$X_a$ and $X_b$ have the same number of used robots* **then**
    **for** $i = 1 \rightarrow f_2(X_a)$ **do**
      The partially matched crossover is applied to $X_{a,i}$ and $X_{b,i}$ for generating offspring permutations $Y_{a,i}$ and $Y_{b,i}$;
    **end**
    $\mathbf{Y} = \{Y_a, Y_b\}$;
  **else**
    **for** $i = 1 \rightarrow \min\{f_2(X_a), f_2(X_b)\}$ **do**
      The partially matched crossover is applied to $X_{a,i}$ and $X_{b,i}$ for generating offspring permutations $Y_{a,i}$ and $Y_{b,i}$;
    **end**
    $\mathbf{Y} = \{Y_a, Y_b\}$;
    **if** $f_2(Y_a) > f_2(Y_b)$ **then**
      Swap $Y_a$ and $Y_b$;
    **end**
    **for** $i = 1 \rightarrow \min\{\binom{f_2(Y_a)}{f_2(Y_b)}, 10\}$ **do**
      Select $f_2(Y_a)$ permutations randomly from $Y_b$ to construct $Y_c$;
      $\mathbf{Y} = \mathbf{Y} \cup Y_c$;
    **end**
    **for** $i = 1 \rightarrow \min\{f_2(X_a), f_2(X_b)\}$ **do**
      Swap $X_{a,i}$ and $X_{b,i}$ purely for generating offspring permuations $Y_{p,i}$ and $Y_{q,i}$;
    **end**
    $\mathbf{Y} = \mathbf{Y} \cup \{Y_p, Y_q\}$;
  **end**
**else**
  Swap mutation operates each permutation of $X_a$ and $X_b$ to generate offspring $Y_a$ and $Y_b$;
  $\mathbf{Y} = \{Y_a, Y_b\}$;
**end**
**return Y**

**Algorithm 3:** The proposed matching mechanism

**Input:** Newly generated individual $Y$, all subproblems with different bound constraints $\{\varepsilon^1, \varepsilon^2 \ldots, \varepsilon^R\}$
**Output:** The selected subproblem $sp_{i^*}$ of population **Pop**

**for** $i = 1 \rightarrow R$ **do**
  $CV^i = \min(\varepsilon_i - \frac{f_2(Y) - K_{lb}}{K_{ub} - K_{lb}}, 0)$;
  **if** $CV^i \neq 0$ **then**
    $CV^i = f_1(sol_i)$
  **end**
**end**
$i^* = \arg\max_{i \in \{1,2,...,R\}} \{CV^1, CV^2, \ldots, CV^R\}$;
**return** $sp_{i^*}$

situations are distinguished. The first situation is that the parent individuals $X_a$ and $X_b$ have the same number of used robots. For each permutation of the parent individuals, the partially matched crossover generates offspring permutations to construct the new offspring individuals **Y**. When the parent individuals $X_a$ and $X_b$ have different numbers of used robots, the new offspring individuals **Y** are constructed by three parts:

1) Similar to the generation method in the situation with the same robot number, the first part of the new offspring individuals is generated based on the partially matched crossover.

2) Let the individual with a larger number of robots in the first part be $Y_b$. The second part of offspring individuals is based on the first part, and it is constructed by selecting $f_2(Y_a)$ permutations from $Y_b$ to construct new individuals. The number of newly constructed individuals in the second part is determined by the binomial coefficient $\binom{f_2(Y_a)}{f_2(Y_b)}$ and hyper-parameter 10.

3) The last part is generated by inheriting the alleles from the two-parent individuals with no implicit mutations.

## 3.5 Matching

After generating a new offspring individual, DMOEA-$\varepsilon$C needs to determine which subproblem is suitable for the individual. This paper designs a subproblem-to-individual matching mechanism to make the best use of information of offspring individuals, details of which are shown in Algorithm 3. In the matching procedure, the subproblem that does not violate constraints and has the maximal makespan objective is selected to compare with the newly generated individual. Since the feasibility rule is implicitly adopted to handle these constrained subproblems, the proposed matching mechanism has a positive impact on the population convergence.

# 4 EXPERIMENTAL SETTING

## 4.1 Data Set and Performance Metrics

Since there is no existing benchmark set for the MO-MPDA problem, we propose a new benchmark set for the MO-MDPA problem to test the performances of algorithms. The settings of instances of the MO-MPDA problem including the number of tasks, task initial demands, the position of tasks, and inherent increment rates follows the settings of the most comprehensive MPDA benchmark set [11]. The ability of each robots in the designed MO-MPDA benchmark is set as the mean ability of all robots in the single-objective MPDA benchmark set [11]. These instances are named by the number of tasks, the position of depot and tasks, and the ratio of the mean inherent increment rates to the robot ability.

Two commonly used performance metrics, i.e., inverted generational distance (IGD) [39] and hypervolume (HV) [40] are employed to evaluate the performance of all compared algorithms in this paper. IGD and HV assess the quality of a nondominated set in terms of convergence and diversity, and their definitions uses the true pareto front. However, the ture pareto front of the MO-MPDA problem is very difficult to obtained due to the complexity of the problem. In this paper, we approximate the true pareto front by selecting non-dominated solutions from all the compared and designed algorithms [15].

Since the scope of a objective value of MO-MPDA (especially the makespan objective) is very large, this paper normalises these objective function values. The IGD and HV values are calculated

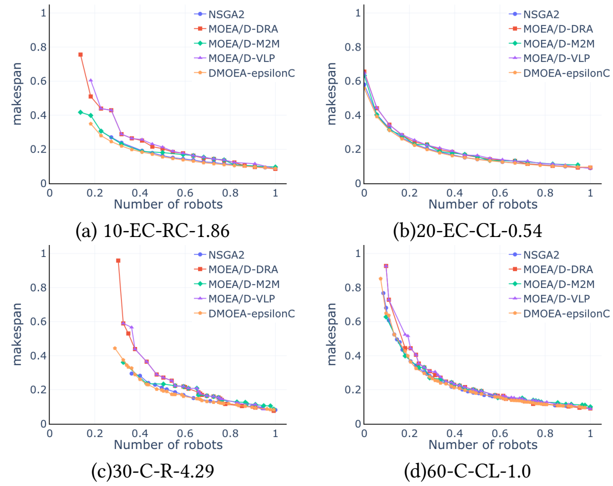


**Figure 4: Final Pareto solutions of different methods for the selected problem instances.One of the Pareto fronts within 20 runs is displayed for illustration in each selected instance.**

based on the normalised objectives of which the range is [0,1], the normalisation methods are shown in Eqs. (18) and (19).

$$f_{1n} = \frac{\ln(\max_{i\in\{1,2,\ldots,N\}} ct_i) - 2\ln(10)}{6\ln(10) - 2\ln(10)} \tag{18}$$

$$f_{2n} = \frac{K - K_{lb}}{K_{ub} - K_{lb}} \tag{19}$$

where $\max_{i\in\{1,2,\ldots,N\}} ct_i$ and $K$ represent the two objective values of MO-MPDA, and $f_{1n}$ and $f_{2n}$ represents the first and second normalised objective values. The large constant $10^6$ and the small constant $10^2$ are viewed as the upper bound and the lower bound of the makespan in nomalisation respectively. The point (1.1, 1.1) is used as the reference point in this paper for calculating the HV metric.

## 4.2 Parameter Settings and Competitor Algorithms

Since the MO-MPDA problem is a novel problem and no existing algorithms can be directly applied for comparison, we compare DMOEA-$\varepsilon$C with the following state-of-the-art methods: NSGA-II [6], MOEA/D-DRA[38], MOEA/D-M2M [23], and MOEA/D-VLP [19].

All the competitor methods and DMOEA-$\epsilon$C for the MO-MPDA problem are implemented based on a Python evolutionary computation framework [10] to keep fair comparisons. The parameter settings of all the methods used in the rest of this paper follow the conventional settings [4, 21, 37]. For each instance, all the algorithms we run 20 times independently.

# 5 EXPERIMENTAL RESULTS AND DISCUSSION

## 5.1 Comparisons with State-of-the-Art Algorithms

The overall results of the compared four state-of-the-art methods and DMOEA-$\epsilon$C method are shown in Tables 1 and 2, including the

**Table 1: Statistical Results of Five Methods over 20 Independent Runs in Terms of IGD**

| Instance | NSGA-II [6] | MOEA/D-DRA [38] | MOEA/D-M2M [23] | MOEA/D-VLP [19] | DMOEA-ϵC |
|---|---|---|---|---|---|
| 10-EC-RC-1.86 | 6.69E-2(8.5E-3)(-) | 7.56E-2(6.7E-3)(-) | 7.42E-2(1.2E-2)(-) | 8.35E-2(8.2E-3)(-) | 4.91E-2(8.5E-3) |
| 10-EC-CL-0.54 | 9.05E-3(3.8E-3)(-) | 2.56E-2(6.3E-3)(-) | 2.03E-2(6.6E-3)(-) | 2.77E-2(2.5E-3)(-) | 5.44E-3(1.4E-3) |
| 20-EC-CL-0.54 | 2.11E-2(2.5E-3)(-) | 3.58E-2(1.9E-3)(-) | 3.14E-2(2.8E-3)(-) | 3.60E-2(1.7E-3)(-) | 1.24E-2(1.6E-3) |
| 30-C-RC-1.86 | 7.38E-2(5.0E-3)(-) | 8.55E-2(3.3E-3)(-) | 8.18E-2(5.0E-3)(-) | 9.32E-2(3.5E-3)(-) | 5.80E-2(3.8E-3) |
| 30-C-R-4.29 | 1.11E-1(8.7E-3)(-) | 1.32E-1(5.6E-3)(-) | 1.16E-1(8.3E-3)(-) | 1.43E-1(4.6E-3)(-) | 9.42E-2(8.2E-3) |
| 40-C-CL-1.86 | 7.14E-2(5.5E-3)(-) | 8.53E-2(2.0E-3)(-) | 7.74E-2(2.8E-3)(-) | 9.01E-2(2.4E-3)(-) | 5.78E-2(3.6E-3) |
| 60-C-CL-1.0 | 4.24E-2(2.6E-3)(-) | 5.25E-2(1.9E-3)(-) | 5.15E-2(3.1E-3)(-) | 5.57E-2(1.9E-3)(-) | 3.69E-2(2.7E-3) |
| 60-C-CL-1.0a | 5.16E-2(2.9E-3)(-) | 6.15E-2(1.8E-3)(-) | 5.93E-2(2.8E-3)(-) | 6.49E-2(1.7E-3)(-) | 4.58E-2(2.7E-3) |
| (+)/(≈)/(-) | 0/0/8 | 0/0/8 | 0/0/8 | 0/0/8 | – |

**Table 2: Statistical Results of Five Methods over 20 Independent Runs in Terms of HV**

| Instance | NSGA-II [6] | MOEA/D-DRA [38] | MOEA/D-M2M [23] | MOEA/D-VLP [19] | DMOEA-ϵC |
|---|---|---|---|---|---|
| 10-EC-RC-1.86 | 8.41E-1(2.5E-2)(-) | 8.24E-1(1.8E-2)(-) | 8.46E-1(3.4E-2)(-) | 8.00E-1(2.5E-2)(-) | 8.84E-1(2.4E-2) |
| 10-EC-CL-0.54 | 1.02E+0(1.2E-3)(-) | 1.00E+0(2.5E-3)(-) | 1.01E+0(1.3E-3)(-) | 9.97E-1(1.9E-3)(-) | 1.02E+0(1.3E-3) |
| 20-EC-CL-0.54 | 1.00E+0(8.4E-4)(-) | 9.89E-1(1.3E-3)(-) | 9.93E-1(1.9E-3)(-) | 9.84E-1(1.3E-3)(-) | 1.01E+0(1.5E-3) |
| 30-C-RC-1.86 | 8.41E-1(1.3E-2)(-) | 8.07E-1(8.4E-3)(-) | 8.29E-1(1.5E-2)(-) | 7.94E-1(1.1E-2)(-) | 8.74E-1(1.0E-2) |
| 30-C-R-4.29 | 7.24E-1(2.1E-2)(-) | 6.86E-1(1.1E-2)(-) | 7.17E-1(2.0E-2)(-) | 6.69E-1(1.0E-2)(-) | 7.52E-1(1.8E-2) |
| 40-C-CL-1.86 | 8.33E-1(1.7E-2)(-) | 7.98E-1(7.4E-3)(-) | 8.22E-1(8.9E-3)(-) | 7.88E-1(9.5E-3)(-) | 8.60E-1(9.6E-3) |
| 60-C-CL-1.0 | 9.08E-1(6.1E-3)(-) | 8.77E-1(6.8E-3)(-) | 8.90E-1(6.1E-3)(-) | 8.69E-1(5.9E-3)(-) | 9.18E-1(7.6E-3) |
| 60-C-CL-1.0a | 8.95E-1(7.0E-3)(-) | 8.64E-1(5.5E-3)(-) | 8.79E-1(6.6E-3)(-) | 8.58E-1(5.7E-3)(-) | 9.05E-1(7.0E-3) |
| (+)/(≈)/(-) | 0/0/8 | 0/0/8 | 0/0/8 | 0/0/8 | – |

**Table 3: Statistical Results of DMOEA-ϵC-TR and DMOEA-ϵC in Terms of IGD**

| Instance | DMOEA-ϵC-TR | DMOEA-ϵC |
|---|---|---|
| 10-EC-RC-1.86 | 4.33E-2(5.9E-3)(+) | 4.91E-2(8.5E-3) |
| 10-EC-CL-0.54 | 5.97E-3(1.8E-3)(≈) | 5.68E-3(1.2E-3) |
| 20-EC-CL-0.54 | 1.64E-2(1.3E-3)(-) | 1.24E-2(1.6E-3) |
| 30-C-RC-1.86 | 6.05E-2(2.3E-3)(-) | 5.80E-2(3.8E-3) |
| 30-C-R-4.29 | 9.87E-2(5.0E-3)(-) | 9.42E-2(8.2E-3) |
| 40-C-CL-1.86 | 6.20E-2(1.7E-3)(-) | 5.78E-2(3.6E-3) |
| 60-C-CL-1.0 | 3.84E-2(1.8E-3)(-) | 3.69E-2(2.7E-3) |
| 60-C-CL-1.0a | 4.73E-2(2.0E-3)(-) | 4.58E-2(2.7E-3) |
| (+)/(≈)/(-) | 1/1/6 | – |

**Table 4: Statistical Results of DMOEA-ϵC-TR and DMOEA-ϵC in Terms of HV**

| Instance | DMOEA-ϵC-TR | DMOEA-ϵC |
|---|---|---|
| 10-EC-RC-1.86 | 8.91E-1(1.4E-2)(≈) | 8.84E-1(2.4E-2) |
| 10-EC-CL-0.54 | 1.02E+0(8.0E-4)(≈) | 1.02E+0(1.4E-3) |
| 20-EC-CL-0.54 | 1.00E+0(7.4E-4)(-) | 1.01E+0(1.5E-3) |
| 30-C-RC-1.86 | 8.69E-1(5.5E-3)(-) | 8.74E-1(1.0E-2) |
| 30-C-R-4.29 | 7.47E-1(1.1E-2)(-) | 7.52E-1(1.8E-2) |
| 40-C-CL-1.86 | 8.54E-1(4.9E-3)(-) | 8.60E-1(9.6E-3) |
| 60-C-CL-1.0 | 9.14E-1(5.3E-3)(-) | 9.18E-1(7.6E-3) |
| 60-C-CL-1.0a | 9.02E-1(5.1E-3)(-) | 9.05E-1(7.0E-3) |
| (+)/(≈)/(-) | 0/2/6 | – |

mean and standard deviation of the performance metrics over all the independent runs. Wilcoxon rank-sum test with a 5% significance level and Bonferroni correction are used to verify the statical results. For each compared algorithm, "(+)", "(≈)", and "(−)" indicate that the compared algorithm performed significantly better, statistically comparable, and significantly worse than the DMOEA-ϵC method, respectively.

From the Tables 1 and 2, it can be observed that DMOEA-ϵC proposed in this paper is significantly better than other comparison methods in terms of IGD and HV. The MO-MPDA planning problem has difficulties such as huge solution space and redundancy representation. These methods do not use the domain knowledge of the MO-MPDA problem during the evolutionary process. In contrast, DMOEA-ϵC effectively utilises the domain knowledge of the MO-MPDA problem. DMOEA-ϵC can effectively focus on the potentially promising solutions by allocating more computational resources. Besides, the subproblems in DMOEA-ϵC can be co-evolved, and they can learn useful knowledge from subproblems with similar robot numbers.

Fig. 4 shows the Pareto fronts of these five methods for the selected problem instances. From Fig. 4, it can be seen that the uniformity of the non-dominated individuals obtained by MOEA/D-εC is obviously better than the other comparison methods. However, all the methods do not perform well for subproblems with tight constraints, in which a small number of robots are used.

Overall, it can be concluded that MOEA/D-εC outperforms the state-of-the-art methods in solving MO-MPDA.

## 5.2 Effectiveness of the Reproduction Mechanism

To verify the effectiveness of the proposed reproduction mechanism for offspring individuals, we adopt the DMOEA-ϵC method with a traditional reproduction mechanism in [13] as a control group, which is denoted as DMOEA-ϵC-TR. The partially matched crossover is applied to each permutation of the parent individuals to construct the new offspring individuals in the DMOEA-ϵC-TR method. Tables 3 and 4 show the comparison results of DMOEA-ϵC-TR and DMOEA-ϵC in terms of IGD and HV.

In view of the statistical results in Tables 3 and 4, DMOEA-ϵC outperforms DMOEA-ϵC-TR on most tested instances. For the instances with a small number of tasks (e.g. 10-EC-RC-1.86 and 10-EC-CL-0.54), the performance of MOEA/D-εC is slightly worse than the performance of MOEA/D-$\varepsilon$C-TR. The reason may be that MOEA/D-$\varepsilon$C-TR, which only uses the partially matched crossover operator, has a strong ability of implicit mutation while generating offspring individuals. The strong implicit mutation leads that MOEA/D-$\varepsilon$C-TR have a good exploration ability, and it makes MOEA/D-$\varepsilon$C-TR easy to jump out of the local optima in the small-scale solution space. However, for the medium-scale and large-scale solution spaces, MOEA/D-$\varepsilon$C that can transmit the information to offspring individuals more effectively gets better performance than MOEA/D-$\varepsilon$C-TR.

## 6 CONCLUSION

The goal of this paper was to solve an emerging and novel MO-MPDA problem from real-world applications. Due to the complex dependencies among robots and tasks, the redundant encoding, and variable-size decision space, the MO-MPDA problem is very challenging. The goal of this paper has been successfully achieved by proposing a MO-MPDA model and designing an elaborate DMOEA-$\epsilon$C method. Specifically, the novel reproduction and matching mechanisms are proposed respectively to promote the effectiveness and efficiency of DMOEA-$\epsilon$C. Experimental results show that DMOEA-$\epsilon$C significantly outperforms the state-of-the-art methods in terms of IGD and HV. The effectiveness and necessity of proposed reproduction mechanism are also validated by the comparison experiments.

There are still some aspects that need to be further investigated to cope with real-world applications: 1) considering the communication delay and loss problem among robots; and 2) implementing a practical multirobot system to verify algorithms for the MPDA problem.

## ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China (2018YFB1308000), in part by the National Outstanding Youth Talents Support Program 61822304, in part by the Basic Science Center Programs of NSFC under Grant 62088101, in part by Beijing Advanced Innovation Center for Intelligent Robots and Systems, in part by Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100), and in part by Shanghai Municipal Commission of Science and Technology Project (19511132101).

## REFERENCES

[1] Mazhar Ansari Ardeh, Yi Mei, and Mengjie Zhang. 2021. A novel multi-task genetic programming approach to uncertain capacitated Arc routing problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 759–767.

[2] Md Asafuddoula, Tapabrata Ray, and Ruhul Sarker. 2014. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation* 19, 3 (2014), 445–460.

[3] Jie Chen, Yuqian Guo, Zhifeng Qiu, and Bin Xin. 2020. Multiagent dynamic task assignment based on forest fire point model. *IEEE Transactions on Automation Science and Engineering* (2020). https://doi.org/Doi:10.1109/TASE.2021.3061757

[4] Jie Chen, Juan Li, and Bin Xin. 2017. Decomposition-based multiobjective evolutionary algorithm with the $\epsilon$-constraint framework. *IEEE Transactions on Evolutionary Computation* 21, 5 (2017), 714–730.

[5] Lu Chen, Bin Xin, and Jie Chen. 2021. Interactive multiobjective evolutionary algorithm based on decomposition and compression. *Science China Information Sciences* 64, 10 (2021), 1–16. https://doi.org/10.1007/s11432-020-3092-y

[6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[7] Saddam Hocine Derrouaoui, Yasser Bouzid, Mohamed Guiatni, and Islam Dib. 2022. A comprehensive review on reconfigurable drones: classification, characteristics, design, and control technologies. *Unmanned Systems* 10, 1 (2022), 3–29. https://doi.org/10.1142/S2301385022300013

[8] Yulong Ding, Bin Xin, Jie Chen, Qingkai Yang, and Hao Fang. 2021. A Unifying Framework for Human-Agent Collaborative Systems-Part II: Design Procedure and Application. *IEEE Transactions on Cybernetics* (2021). https://doi.org/10.1109/TCYB.2021.3086073

[9] Xin Du, Jia Zhang, Bin Xin, Yulong Ding, Zhihong Peng, and Lihua Dou. 2018. Market-based task assignment for multi-point dynamic aggregation tasks. In *Proceedings of International Conference on Computers and Industrial Engineering*. Auckland, New zealand, 1667–1678.

[10] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: evolutionary algorithms made easy. *Journal of Machine Learning Research* 13 (2012), 2171–2175.

[11] Guanqiang Gao, Yi Mei, Yahui Jia, Will N Browne, and Bin Xin. 2021. Adaptive coordination ant colony optimization for multipoint dynamic aggregation. *IEEE Transactions on Cybernetics* (2021). https://doi.org/Doi:10.1109/TCYB.2020.3042511

[12] Guanqiang Gao, Yi Mei, Bin Xin, Ya-Hui Jia, and Will N Browne. 2021. Automated Coordination Strategy Design Using Genetic Programming for Dynamic Multipoint Dynamic Aggregation. *IEEE Transactions on Cybernetics* (2021). https://doi.org/10.1109/TCYB.2021.3080044

[13] Guanqiang Gao, Mei Yi, Xin Bin, Jia Ya-Hui, and Browne Will. 2020. A memetic algorithm for the task allocation problem on multi-robot multi-point dynamic aggregation missions. In *Proceedings of the 2020 IEEE World Congress on Computational Intelligence*. IEEE, 201–208.

[14] Rong Hao, Jia Zhang, Bin Xin, Chen Chen, and Lihua Dou. 2018. A hybrid differential evolution and estimation of distribution algorithm for the multi-point dynamic aggregation problem. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 251–252.

[15] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. 2014. Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems. In *Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*. IEEE, 170–177.

[16] Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. 2009. An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research* 195, 3 (2009), 761–769.

[17] Sašo Karakatič and Vili Podgorelec. 2015. A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing* 27 (2015), 519–532.

[18] Shahbaz Khan, Muhammad Tufail, Muhammad Tahir Khan, Zubair Ahmad, Javaid Iqbal, and Arsalan Wasim. 2022. A novel framework for multiple ground target detection, recognition and inspection in presicion agriculture applications using a UAV. *Unmanned Systems* 10, 1 (2022), 45–56. https://doi.org/10.1142/S2301385022500029

[19] Hui Li, Kalyanmoy Deb, and Qingfu Zhang. 2019. Variable-length Pareto optimization via decomposition-based evolutionary multiobjective algorithm. *IEEE Transactions on Evolutionary Computation* 23, 6 (2019), 987–999.

[20] Hui Li and Qingfu Zhang. 2008. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation* 13, 2 (2008), 284–302.

[21] Juan Li, Jie Li, Panos M Pardalos, and Chengwei Yang. 2020. DMaOEA-$\varepsilon$C: decomposition-based many-objective evolutionary algorithm with the $\varepsilon$-constraint framework. *Information Sciences* 537 (2020), 203–226.

[22] Juan Li, Bin Xin, Jie Chen, and Panos M Pardalos. 2018. Noise-Tolerant Techniques for Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE transactions on cybernetics* 50, 5 (2018), 2274–2287.

[23] HaiLin Liu, Fangqing Gu, and Qingfu Zhang. 2013. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation* 18, 3 (2013), 450–455.

[24] Jianyu Long, Zhenzhong Sun, and Panos M Pardalos. 2019. A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem. *Information Sciences* 478 (2019), 40–61.

[25] Sai Lu, Bin Xin, Lihua Dou, and Ling Wang. 2018. A Multi-Model Estimation of Distribution Algorithm for Agent Routing Problem in Multi-Point Dynamic Task. In *the Proceedings of the 2018 Chinese Control Conference*. IEEE, 2468–2473.

[26] Ferdinando Pezzella, Gianluca Morganti, and Giampiero Ciaschetti. 2008. A genetic algorithm for the flexible job-shop scheduling problem. *Computers and Operations Research* 35, 10 (2008), 3202–3212.

[27] Jeanyves Potvin and Samy Bengio. 1996. The vehicle routing problem with time windows part II: genetic search. *Informs Journal on Computing* 8, 2 (1996), 165–172.

[28] Georgy Skorobogatov, Cristina Barrado Muxí, and Esther Salamí San Juan. 2020. Multiple UAV systems: a survey. *Unmanned System* 8, 2 (2020), 149–169.

[29] Sufal Chandra Swar, Suresh Manickam, David Casbeer, Krishna Kalyanam, and Swaroop Darbha. 2022. Optimal autonomous persuit of an intruder on a grid aided by local node and edge sensors. *Unmanned Systems* 10, 1 (2022), 93–108. https://doi.org/10.1142/S2301385022500054

[30] KuangYee Teng and Jayantha Katupitiya. 2014. Market-based task assignment strategies for multi-agent systems deployed for bushfire fighting. In *Proceedings of the 11th IEEE International Conference on Control and Automation*. IEEE, 25–31.

[31] Paolo Toth and Daniele Vigo. 2014. *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics, Pennsylvania, USA.

[32] Jiahai Wang, Taiyao Weng, and Qingfu Zhang. 2018. A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows. *IEEE Transactions on Cybernetics* 49, 7 (2018), 2467–2478.

[33] Shaolin Wang, Yi Mei, and Mengjie Zhang. 2021. Two-stage multi-objective genetic programming with archive for uncertain capacitated arc routing problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 287–295.

[34] Yingbo Xie, Shengxiang Yang, and Ding Wang. 2022. Dynamic Transfer Reference Point Oriented MOEA/D Involving Local Objective-Space Knowledge. *IEEE Transactions on Evolutionary Computation* (2022). https://doi.org/10.1109/TEVC.2022.3140265

[35] Bin Xin, Shiqing Liu, Zhihong Peng, and Guanqiang Gao. 2018. An estimation of distribution algorithm for multi-robot multi-point dynamic aggregation problem. In *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 775–780.

[36] Bin Xin, Yangguang Zhu, Yulong Ding, and Guanqiang Gao. 2016. Coordinated motion planning of multiple robots in multi-point dynamic aggregation task. In *Proceedings of the 12th IEEE International Conference on Control and Automation*. IEEE, 933–938.

[37] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (2007), 712–731.

[38] Qingfu Zhang, Wudong Liu, and Hui Li. 2009. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *Proceedings of the 2009 IEEE on Evolutionary Computation*. IEEE, 203–208.

[39] Aimin Zhou, Qingfu Zhang, Yaochu Jin, Edward Tsang, and Tatsuya Okabe. 2005. A model-based evolutionary algorithm for bi-objective optimization. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, Vol. 3. IEEE, 2568–2575.

[40] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999), 257–271.